

Figure 1. Presentation Generation Process

```
<CardLayoutStyle Name=name . . . >
  <Options size=card_size . . . > . . . </Options>
  . . .
  <BackgroundList>
    <Background name=name Func=function_name>
      <object property= . . . ></object>
      . . .
    </Background>
    <Background name=name Func=function_name>
      <object property= . . . ></object>
      . . .
    </Background>
    . . .
  </BackgroundList> . . .
  <ForegroundList>
    <Foreground name=name Func=function_name>
      <object property= . . . ></object>
      . . .
    </Foreground>
    <Foreground name=name Func=function_name>
      <object property= . . . ></object>
      . . .
    </Foreground>
    . . .
  </ForegroundList>
  <CardSeq name=name Context=context Type=type Content=content>
    <Background StyleRef=background_name>
  </CardSeq>
  <CardSeq name=name Context=context Type=type Content=content>
    <Background StyleRef=background_name>
  </CardSeq>
  . . .
  <Card name=name Context=context Type=type Content=content>
    <Foreground StyleRef=background_name></Foreground>
  </Card>
  <Card name=name Context=context Type=type Content=content>
    <Foreground StyleRef=background_name></Foreground>
  </Card>
  . . .
<CardLayoutStyle>
```

Figure 2. Card Layout Style Specification

```

<CardLayoutStyle Name="name . . .>
  <Options size=card_size . . .> . . . </Options>
  . . .
  <BackgroundList>
    <Background name=name Func=function_name>
      <object property= . . .></object>
    . . .
    </Background>
    <Background name=name Func=function_name>
      <object property= . . .></object>
    . . .
    </Background>
  . . .
</BackgroundList> . . .
<ForegroundList>
  <Foreground name="TextByGfxForeground" Func="TextByGfxForeground">
    <Object Name="TextByGfxTitle"
      Coords="180,1395,1150,1305"
      Type="Field"
      Func="TitleText"
      Context="Card,CardX, CardTitle"
      FontSize="15"
      FontStyle="Bold"
      FontColor="blue"
      Content="CardTitle"></Object>
    . . .
    <Object Name="GfxContent"
      Coords="357 94 881 693"
      Type="Graphics"
      Func="GfxContent"
      Context="Card,CardX, AnyDocX, Figure, Graphic"
      Content="Graphic"
      AIU="Card, CardX, AnyDocX, Figure, AIUDoc, ROI"></Object>
    . . .
  </Foreground>
  . . .
</ForegroundList>
<CardSeq name="TextByGfxSeq" Type="TextByGfx"
  Context="PlantLevel,CardSeq" Content="">
  <Background StyleRef="NormalBackground">
</CardSeq>
. . .
<Card name="TextByGfxCard" Type="TextByGfx"
  Context="PlantLevel, CardSeq, Card" Content="">
  <Foreground StyleRef="TextByGfxForeground"></Foreground>
</Card>
. . .
<CardLayoutStyle>

```

Figure 3. Example of CardTitle Layout Style Specifications

162260 * 2007-07-16

```
<MACRO_Declaration>
  <ObjectGroup name="project_resource">
    MACRO_Resource(stylename, type, filename)</>
    MACRO_Resource(stylename, type, filename)</>
    ...
  </ObjectGroup>
  MACRO_name(stylename, parameters) </>
  MACRO_name(stylename, parameters) </>
  ...
</MACRO_Declaration>
<ContentMapping>
  <ElementRule Context="doc_tags">
    <DefineVar name="..."/>
    <DefineVar name="..."/>
    ...
    <CaseExpr key="..."/>
    <CaseClause matchvalue="..."/>
    <AddObject otype="..." stylename="..."/>
    <ProcessGrove command="..."/></>
    ...
    <AddGroup groupname="..."/></>
    ...
  </AddObject>
  </CaseClause>

  <CaseClause matchvalue="..."/>
  <AddObject otype="..." stylename="..."/>
  <ProcessGrove command="..."/></>
  ...
  <AddGroup groupname="..."/></>
  ...
  </AddObject>
  </CaseClause>
  </CaseExpr>
  </ElementRule>

  <ElementRule ...> ... </>
  ...
</ContentMapping>
```

Figure 4. Card-Based Presentation Specification

```

<MACRO_Declaration>
  ...
  MACRO_TitleField(TextByGfxTitle, "180,1395,1150, 1305, "$CardTitle_RTF$, 15", "bold", "blue")
  ...
</MACRO_Declaration>

<ContentMapping>
  ...
  <ElementRule context="(Card CardX)">
    <ProcessGrove
      command="process-matching-children CardTitle">
    <ProcessGrove
      command="process-matching-children ANYDOCX">
    </ElementRule>

    <ElementRule context="(Card CardX CardTitle)">
      <DefineVar name="$CardTitle_RTF" forvalue="(RTF-TitleGen (current node))"></>

      <CaseExpr key= "(CardType (current-node) )">
        <CaseClause matchvalue= "GfxonlyCard">
          <AddObject otype="Field" stylename="GfxOnlyTitle"></>
        </CaseClause>
        <CaseClause matchvalue= "TextByGfxtitle">
          <AddObject otype="Field" stylename="TextByGfxTitle"></>
        </CaseClause>
        <CaseClause matchvalue= "else">
          <AddObject otype="Field" stylename="TextByGfxTitle"></>
        </CaseClasuse>
      </CaseExpr>
    </ElementRule>
    <ElementRule> . . . </>
  ...
</ContentMapping>

```

Figure 5. Example of Card-Based Presentation Specification for CardTitle

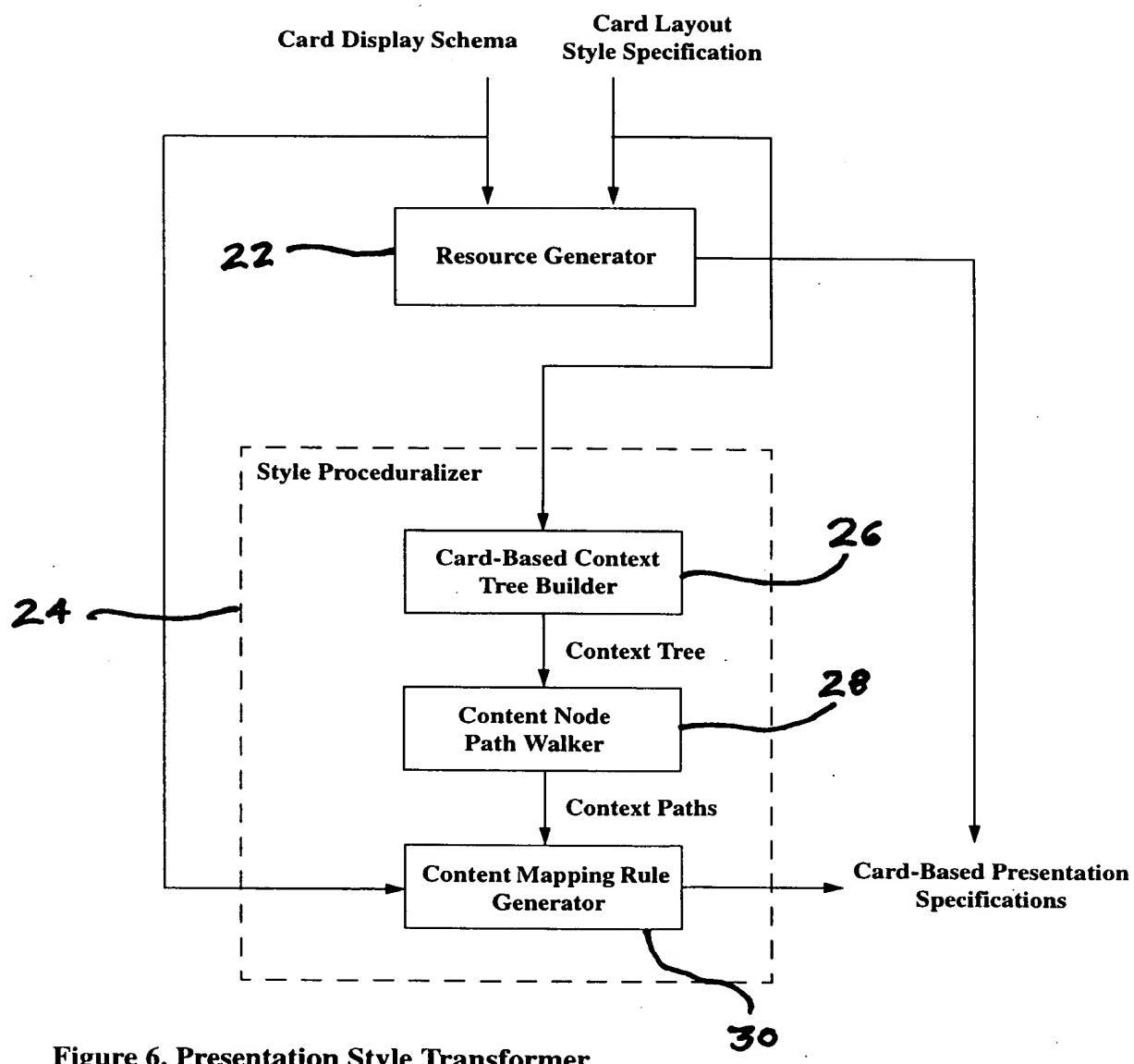


Figure 6. Presentation Style Transformer

```
<DisplaySchema Objectype="Field" Name="TextByGfxTitle" >
  <Argument name="StyleName" type="Required"
    AvailValue="#ANY_UNIQUE_NAME">
  <Argument name="vertices" type="Required"
    Availvalue="#ANY_2POINTS">
  <Argument name="TitleRTF" type="Optional"
    DefaultValue=$CARDTitle_RTF$ >
  <Argument name="FontSize" type="Optional"
    DefaultValue="15" AvailValue="#Number">
  <Argument name="Fontstyle" type="Optional"
    DefaultValue="Regular" AvailValue="Bold,Regular, Italic">
  <Argument name="FontColor" type="Optional"
    DefaultValue="Black" AvailValue="#ANY_Color_Name or RGBStoke">
</DisplaySchema>
```

Figure 7. Display Schema for Generating MACRO_ TitleField

```
<DisplaySchema Objecttype= object_type Name= macro_name or variable_name>
  <Argument name= . . . type= . . .
    DefaultValue= . . . AvailValue= . . . >
  <Argument name= . . . type= . . .
    DefaultValue= . . . AvailValue= . . . >
  <Argument name= . . . type= . . .
    DefaultValue= . . . AvailValue= . . . >
  . . .
</DisplaySchema>
```

Figure 8. Display Schema for Generating Resource Declarations or Variable Definition

```
<DisplaySchema Objecttype='DefineVar' Name="$CardTitle_RTF$" >
  <Argument name="RTF-Gen" type="Required"
    AvailValue="#ANY-DSSL-FUNC-NANME">
    <Argument name="context" type="Optional"
      DefaultValue="(Card Title)">
  </DisplaySchema>
```

Figure 9. Display Schema for Generating Content Variable Definitions

```
<DisplaySchema Objecttype="Resource" MacroName="MACRO_Resource">
  <Argument name="Format" type="Optional"
    DefaultValue="icon" AvailValue="icon,bitmap,sharedscript, menubar">
  <Argument name="StyleName" type="Required"
    AvailValue="#ANY_UNIQUE_NAME">
  <Argument name="InstanceName" type="Optional"
    DefaultValue="" AvailValue="#ANY_STRING">
  <Argument name="FileName" type="Required"
    AvailValue="#ANY_STRINGr">
</DisplaySchema>
```

Figure 10. Display Schema for Generating all Primitive MACRO_ Resources

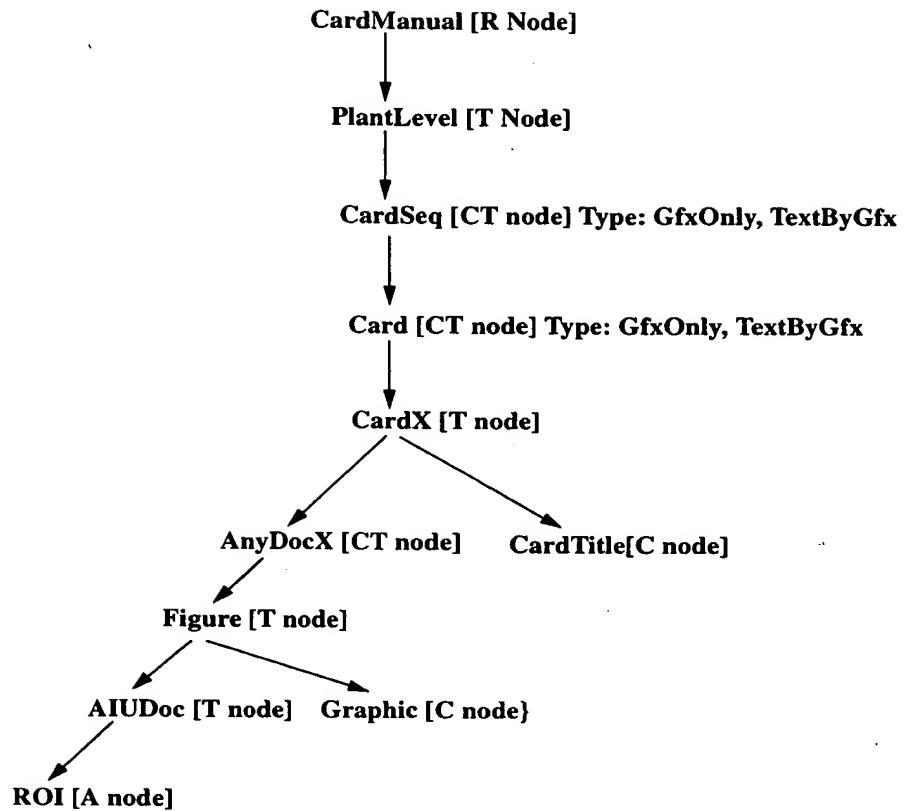


Figure 11. Example of Card-Based Context Tree

1. **CardManual, PlantLevel, CardSeq**
2. **CardManual, PlantLevel, CardSeq, Card**
3. **CardManual, PlantLevel, CardSeq, Card, CardX, CardTitle**
4. **CardManual, PlantLevel, CardSeq, Card, CardX, AnyDocX**
5. **CardManual, PlantLevel, CardSeq, Card, CardX, AnyDocX, Figure, Graphic**
6. **CardManual, PlantLevel, CardSeq, Card, CardX, AnyDocX, Figure, AIUDoc, ROI**

Figure 12. Example of Card-Based Context Paths for C, CT and A Nodes

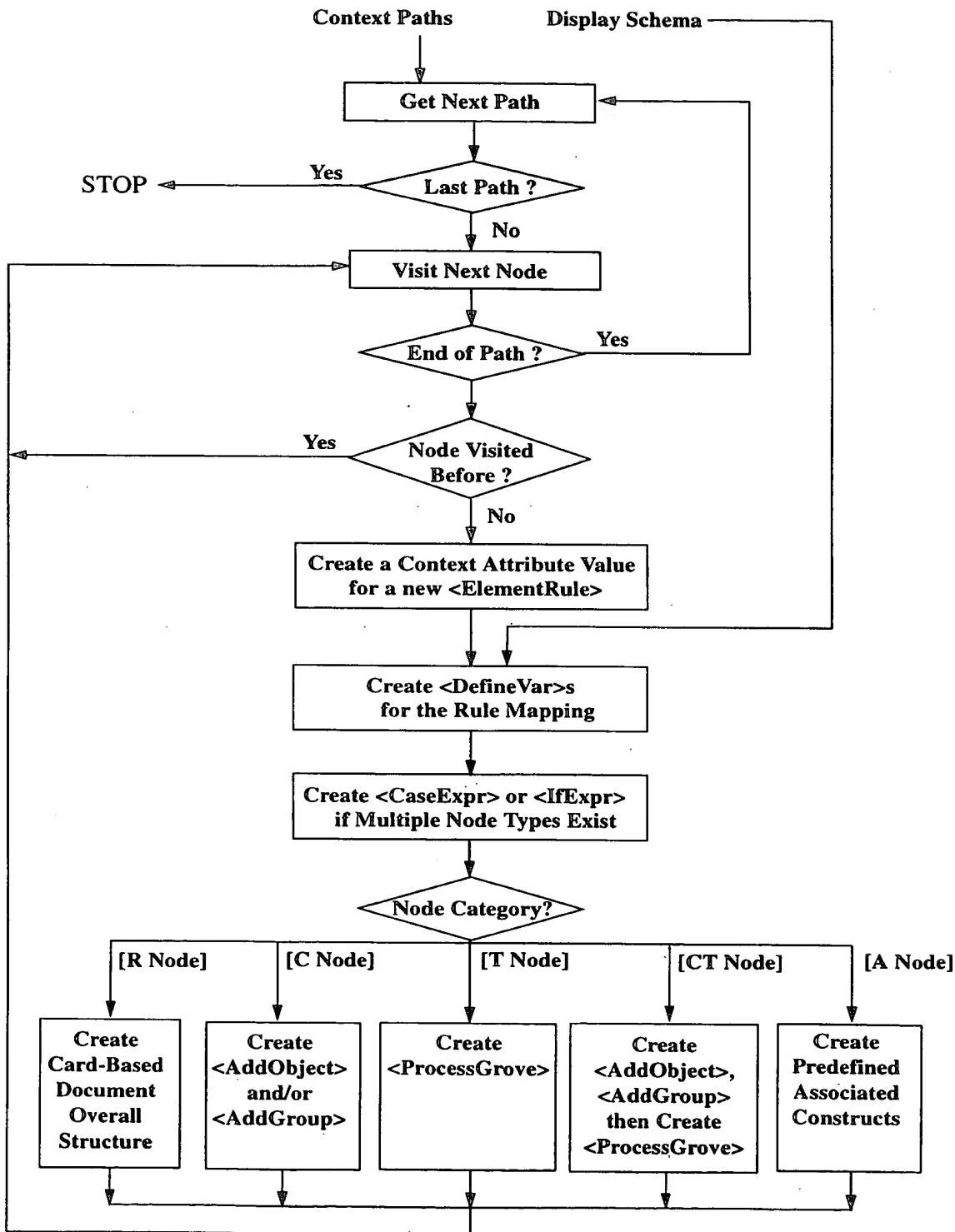


Figure 13. Content Mapping Rule Generation Process

```

(A) Transition Node: CardX
<ElementRule context="(Card CardX)">
<ProcessGrove
  command="process-matching-children CARDTITLE"></>
<ProcessGrove
  command="process-matching-children ANYDOCX"></>
</ElementRule>

(B) Content Node: CardTitle (See Figure 5)

(C) Content Transition Node: CardSeq
<ElementRule context="(PlantLevel CardSeq)">
<DefineVar name="$Page_currentObject$"
  strValue="(currentObjectGen (current-node))" > </>
<DefineVar ...> </>
...
<CaseExpr key="(cardSeqType (current-node))">

  <CaseClause matchval="GfxOnly">
    <AddObject otype=Background styleName="GfxOnlyBackground">
    <ProcessGrove
      command="process-matching-children CARD"></>
    </AddObject>
  </CaseClause>

  <CaseClause matchval="TextByGfx">
    <AddObject otype=Background styleName="NormalBackground">
    <AddObject otype=Button styleName="Top"></>
    <AddObject otype=Button styleName="First"></>
    <AddObject otype=Button styleName="Previous"></>
    <AddObject otype=Button styleName="Next"></>
    <AddObject otype=Button styleName="Last"></>
    <AddObject otype=Button styleName="Audio"></>
    <AddObject otype=Button styleName="Animate"></>
  ...
    <ProcessGrove
      command="process-matching-children CARD"></>
  </CaseClause>

  <CaseClause matchval="else">
  ...
</CaseClause>

</CaseExpr>
</ElementRule>

```

Figure 14. Example of Generated CPS for C, CT and T Nodes

```

(D) ROOT NODE: CardManual
<ElementRule context="CardManual">
<AddObject otype=CardManualLayout styleName=" ">
<DefineVar name="$BookTOC$" strValue="(TOC-Gen (current-node))" > </>
<AddObject otype=ToolBookManual styleName="IMIS_ToolBook">
<AddGroup groupName="Project _Resources" > </>
<AddObject otype=MainViewer styleName="MainViewer">
<ProcessGrove
  command="process-matching-children PLANTLEVEL"></>
<AddGroup groupName="IMIS_MiscViewersContent" > </>
</AddObject>
<AddGroup groupName="IMIS_MiscViewers" > </>
</AddObject>
</AddObject>
</ElementRule>

(E) Associated Node: ROI
<ElementRule context="(Card CardX AnyDocX Figure AIUDoc ROI)">
<DefineVar name="$AiuName$"
  strValue="(getattr NAME (current-node))" > </>

<CaseExpr key="(cardType (current-node))">

<CaseClause matchval="GfxOnly" > ...</CaseClause>
<CaseClause matchval="TextByGfx" >
<CaseExpr key="(ROI-TYPE (current-node))" >
<CaseClause matchval="polygon" >
<AddObject otype=Polygon styleName="GfxContent-AIU-Poly"></>
</>
<CaseClause matchval="callout" > ..</>
<CaseClause matchval="NamedCallout" > ..</>
...
</CaseExpr>
</CaseClause>
<CaseClause matchval="else" >
<EmptySosofo>
</CaseClause>

</CaseExpr>
</ElementRule>

```

Figure 15. Example of Generated CPS for R and A Nodes

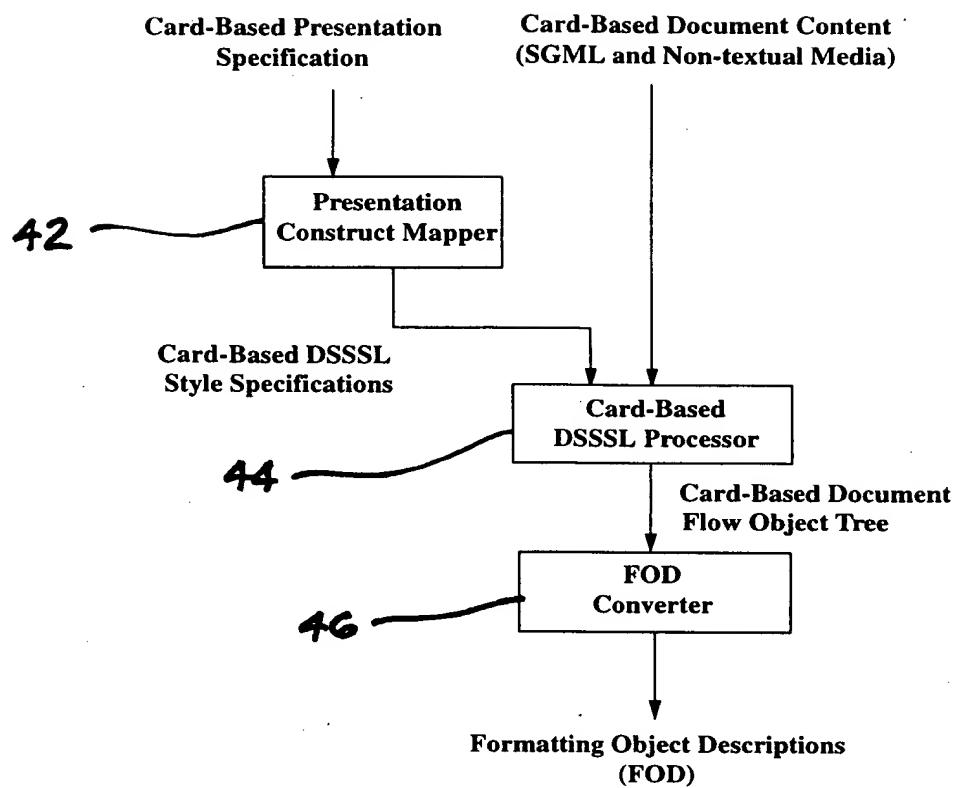


Figure 16. Hypermedia Document Presentation Generator

```
(element (Card CardX CardTitle)
  (sosofo-append
    (makeFODfo FOtype: "DefineVar"
      FOname: "$CardTitle_RTF$"
      FOinstruction: (RTF-TitleGen (current Node)))
    (case (cardType (current-node))
      ("GfxOnly"
        (makeFODfo FOtype: "StartTag"
          FOname: "Field"
          FOinstruction: (AttributeGen GfxOnlyTitle)))
      ("TextByGfx"
        (makeFODfo FOtype: "StartTag"
          FOname: "Field"
          FOinstruction: (AttributeGen TextByGfxTitle)))
      (else
        (makeFODfo FOtype: "StartTag"
          FOname: "Field"
          FOinstruction: (AttributeGen TextByGfxTitle)))
    )
  )
)
```

Figure 17. Generated Card-Based DSSSL Style for CardTitle

```

(define (AttributeGen type)
(case type
  (("TextByGfxTitle")
   (string-append
    " FontSize=\"15\" "
    " FontStyle=\"bold\" "
    " FontColor=\"0,0,255\" "
    " RGBFILL=\"255,255,255\" "
    " TRANSPARENT=\"true\" "
    " VERTICES=\"180,1395,1150,1305\" "
    " RICHTEXT=\"$CardTitle_RTF$\" "
    " FONTFACE=\"Times New Roman\" "
    " ACTIVATED=\"true\" "
    " BASELINES=\"false\" "
    " BORDERSTYLE=\"none\" "
    " FIELDTYPE=\"wordWrap\" "
    " SCROLL=\"0\" "
    " SPACING=\"1\" "
    " INDENTS=\"0,0,0\" "
    " TABSPACING=\"360\" "
    " TABTYPE=\"left\" "
    " TEXTALIGNMENT=\"left\" "
    " NAME=\"TitleField\""))
  (("GfxOnlyTitle")
   ...))
  ...)

(define (makeFODfo #! key Fotype FOname FOinstruction)
(case Fotype
  (("StartTag")
   (make FOD-Instruction
         data: (string-append "<" FOname FOinstruction "> </>")))
  (("DefineVar")
   (make FOD-Instruction
         data: (string-append "<DefineVar name=" FOname "value=" FOinstruction ">")))
  (("EndTag")
   (make FOD-instruction
         data: (string-append "</>")))
  ...
  (else
   (make FOD-instruction
         data: (string-append "newFlowObjectType")))))
)

```

Figure 18. Card-Based DSSSL Style Functions for CardTitle

0934032655 09322555

```
<CardFormattingBook property= . . . >
  <Resource property= . . . > . . . </Resource>
  <Resource property= . . . > . . . </Resource>
  . . .
  <MainViewer property= . . . >
    <StartBkgnd property= . . . >
      <Object property= . . . >
      . . .
      <StartPage property= . . . > . . . </StartPage>
      <NextPage property= . . . >
        <Field Name= "TitleField"
          TextAlign="left"
          TabType="left"
          TabSpacing="360"
          Scroll="0"
          FieldType="wordwrap"
          BorderStyle="none"
          BaseLines="false"
          Activated="true"
          Fontface="Time New Roman"
          Vertice="180,1395,1150,1305"
          Transparent="True"
          RichText=
            "<\rtf1\ansi \deff5\deftab360\deflang1033
            {\fonttbl
            {\f1\froman\fcharset2\fprq2 Symbol;}
            {\f2\ftech Marlett;}
            {\f4\froman\fcharset0\fprq2 Times New Roman;}
            {\f5\fswiss\fcharset0\fprq2 Arial;}
            {\f10\fswiss\fcharset0\fprq2 System;}}
            {\colortbl1\red0\green0\blue0;\red0\green0\blue255;
            \red255\green0\blue0;\red255\green255\blue255;}
            \pard\plain \f4\fs20 {\b\i\f5\fs36\cfl CO2 Purge System 1}
            >
            "
          RGBFill="255,255,255"
          fontsize="15"
          fontStyle="bold"
          rgbStroke="0, 0, 255"
        >
      </NextPage>
      <NextPage property= . . . > . . . </NextPage>
      . . .
    </StartBkgnd>
    <NextBkgnd property= . . . > . . . </NextBkgnd>
    . . .
  </MainViewer>
  <Viewer property= . . . > . . . </Viewer>
  . . .
</CardFormattingBook>
```

Figure 19. CardTile Formatting Object Description

```
<CardFormattingBook property= . . . >
  <Resource property= . . . > . . . </Resource>
  <Resource property= . . . > . . . </Resource>
  . . .
  <MainViewer property= . . . >
    <StartBkgnd property= . . . >
      <Object property= . . . >
      . . .
      <StartPage property= . . . >
        <Object property= . . . >
        . . .
        </StartPage>
        <NextPage property= . . . >
          <Object property= . . . >
          . . .
          </NextPage>
        <NextPage property= . . . >
          <Object property= . . . >
          . . .
        </NextPage>
        . . .
      </StartBkgnd>
      <NextBkgnd property= . . . > . . . </NextBkgnd>
      . . .
    </MainViewer>
    <Viewer property= . . . > . . . </Viewer>
    <Viewer property= . . . > . . . </Viewer>
    . . .
  </CardFormattingBook>
```

Figure 20. Card-Based Formatting Object Description